CLAIMS

1. A media processing system interface comprising:

an input, coupled to a source to access content from the source in response to requests for said content; and

two or more outputs, each coupled to a media processing subsystem, wherein the interface routes at least a subset of the received media content to individual ones of the two or more outputs based, at least in part, on the media type of the subset.

2. A media processing system interface according to claim 1, wherein the media processing system interface is a software object, exposed from an operating system on computing system implementing the media processing system.

3. A media processing system interface according to claim 1, wherein the interface receives requests from each of the one or more media processing subsystems, and initiates seeks into the source for the requested content.

4. A media processing system interface according to claim 1, wherein the interface receives requests from each of the one or more media processing subsystems for particular source content, ignores the requests received from all but one of the media processing subsystems.

5. A media processing system interface according to claim 1, wherein a source processing chain comprising each of the media processing subsystems coupled through the interface to the source is removed from an active filter graph upon completion by each of the media processing subsystems.

6. A media processing system interface according to claim 5, wherein a first of the media processing subsystems instructs a second of the media processing subsystems that it no longer requires content from the source, and the second media processing subsystem informs a render engine controlling the filter graph to remove the source chain when it no longer requires content from the source.

7. A media processing system interface according to claim 6, wherein the render engine determines whether the source chain may be required subsequently in this, or another media processing project and, if so, caches the source chain for later retrieval and integration in a processing project.

8. A media processing system interface according to claim 5, wherein a first of the media processing subsystems request to remove the source processing chain is ignored, wherein only the second of the media processing subsystems can effectively request the source chain to be removed from the active processing project.

9. A media processing system interface according to claim 5, wherein a render engine controlling the project determines whether at least a subset of the source filter chain will subsequently be required and, if so, caches the source chain in local memory for subsequent retrieval and integration into a processing project.

10. A media processing system according to claim 1, wherein the interface receives media content from a source and parses the received content into its disparate content types.

11. A media processing system according to claim 10, wherein each of the media processing subsystems take one type of the parsed media content for subsequent processing.

12. A media processing system according to claim 10, wherein the disparate content types include audio media content and video media content.

13. A media processing system interface according to claim 1, wherein the media processing subsystems include a processing chain of software objects which manipulate the content retrieved from the source in some fashion.

14. A media processing system interface according to claim 1, wherein the interface is a filter in a media processing filter graph.

15. A media processing system interface according to claim 1, the input is a software object, exposed from the operating system of a computer implementing a media processing system, and implemented as an input pin.

16. A media processing system interface according to claim 1, wherein each of the outputs are software objects, exposed from the operating system of a computer implementing a media processing system, and implemented as instances of an output pin.

17. A media processing system interface according to claim 1, wherein the interface is a parser object, implemented in a filter graph by render engine to enable multiple processing subsystems to access and receive content from a single instance of a source.

18. A storage medium comprising a plurality of executable instructions which, when executed, implement a media processing system interface according to claim 1.

19. A computer system comprising:

a storage medium having stored thereon a plurality of executable instructions; and

an execution unit, coupled to the storage medium, to execute at least a subset of the plurality of executable instructions to implement a media processing system interface according to claim 1.

20. A filter graph implemented within a media processing system, the filter graph comprising:

a video processing subsystem to process video content;

an audio processing subsystem to process audio content; and

a parser object, coupling one or more of the video processing subsystem and the audio processing subsystem to a single instance of a multimedia source, to selectively provide the audio subsystem and video subsystem with requested audio content and video content, respectively.

21. A filter graph according to claim 20, wherein the parser is comprised of at least one input, coupled to the source, and two outputs, one each coupled to the video processing subsystem and the audio processing subsystem.

22. A filter graph according to claim 20, wherein the parser object receives request for content from each of audio processing subsystem and the video processing subsystem and serializes such requests, processing them in chronological order.

23. A filter graph according to claim 20, wherein the parser object receives requests for content from each of the audio processing subsystem and the video processing subsystem and ignores requests received on all but a selected one of such audio processing subsystem or video processing subsystem.

24. A filter graph according to claim 23, wherein the video processing subsystem is the selected one from which requests for content are acted upon, while requests from the audio processing subsystem are ignored.

25. A filter graph according to claim 20, wherein the parser object receives indications from one or more of the audio processing subsystem and/or the video processing subsystem that the source is no longer required and, upon verifying that neither processing subsystem requires further content from the source before a source filter chain including the parser object and both media processing subsystems may be removed from the filter graph.

26. A filter graph according to claim 25, wherein the parser object informs a render engine that the source filter chain is no longer required, whereupon the render engine may remove the source filter chain from the filter graph.

27. A filter graph according to claim 26, wherein the render engine determines whether the source filter chain will be required subsequently in this or another filter graph and, if so, caches the entire source filter chain for later integration in the an appropriate filter graph.

28. A filter graph according to claim 20, wherein the parser object only responds to an indication that the source is no longer required from a select one of the media processing subsystems.

29. A filter graph according to claim 28, wherein the select media processing subsystem is the video processing subsystem.

30. A filter graph according to claim 29, wherein the parser object ignores indications that the source is no longer required from the audio processing subsystem.

31. A filter graph according to claim 29, wherein an element of the video processing subsystem checks with other media processing subsystems coupled to the parser object to determine whether they, too, no longer require content from the source before instructing the parser object that the source is no longer required.

32. A filter graph according to claim 29, wherein the parser ensures that both subsystems no longer require content from the source before unloading source filter strings comprising the audio subsystem and the video subsystem.

33. A filter graph according to claim 20, wherein the parser receives indications from each of the audio and video subsystems requesting content and resolves such requests based, at least in part, on a priority of when such content is required in support of filter graph execution.

34. A computing system comprising:

a storage medium having stored therein a plurality of executable instructions; and

an execution unit, coupled to the storage medium, to execute at least a subset of the plurality of executable instructions to implement a filter graph according to claim 20.

35. A storage medium comprising a plurality of executable instructions which, when executed, implement a media processing system including a parser object, to couple one or more media processing subsystems to a single instance of a source to provide each of the coupled media processing subsystems with requested content from the source.

36. A storage medium according to claim 35, wherein each of the media processing subsystems process media content of a particular type, and wherein the parser object parses out that type of media content from the source for provision to each of the media processing subsystems.

37. A storage medium according to claim 35, wherein the parser object receives multiple requests for content from the media processing subsystems and serializes such requests, processing them in chronological order.

38. A storage medium according to claim 35, wherein the parser object receives requests for content from each of the one or more media processing subsystems, but only acts on requests received from a single media processing subsystem.

39. A storage medium according to claim 35, wherein the parser object receives indications from each of the one or more media processing subsystems that content from the source is no longer required, but ignores all but those indications received from a select media processing subsystem.

40. A storage medium according to claim 35, wherein the parser object receives an indication from one of a plurality of coupled media processing subsystems that content from the source is no longer required, and verifies that none of the remaining plurality of coupled media processing subsystems require further content before releasing the source.